GWD-C
SEC S3A-RG

Shawn Mullen, IBM
Matt Crawford, FNAL
Markus Lorch, VT
Dane Skow, FNAL

June 2003
Web Site: http://www.ppdg.net/pa/ppdg-pa/siteaa/GGF-SiteAAA-RG/

# Grid Authentication Authorization and Accounting Requirements Research Document

The purpose of this research paper is to collect and codify the requirements of existing grid resource sites with respect to the acceptance of grid credentials for access to their services. Where those requirements are non-uniform, or even mutually exclusive, the group will strive to recommend hooks which grid toolkits or applications should provide for the sites to insert their own implementations of their requirements.

This research paper is an informational or community practices GGF document which grid application and library coders can use as a reference guide, and suggestions for future development work in GGF working groups.

Index



## Chapter 1 Site Authentication Requirements

### 1.1 Terminology and definitions

The following terms are used in this document as described here.

1.1.1
"User secrets" refers to values intended to be known only by the user, known by the user and an authentication infrastructure, or known only to an authentication infrastructure and employed on the user's behalf after the user has authenticated with some other secret(s).

1.1.2
To sidestep such questions as whether "a day" means eight hours or 24 hours and just how long a month is, we will deal in seconds but not quibble over implementation variances at the 10% or 20% level.

1.1.3
Credentials are assumed to have lifetimes which bound their period of validity. "Long-lived" credentials have lifetimes of 1,000,000 seconds (1 megasecond or 1 Ms) or more. "Short-lived" credentials have lifetimes of 100,000 seconds (0.1 Ms) or less. Lifetimes between those limits are "intermediate." The terms long-lived and short-lived may also be applied to the secrets employed by a user to acquire credentials, although the only short-lived user secrets known to be commonly employed are one-time (or "single-use") authenticators.

(Conversions: 0.1 Ms is a bit more than a day; 1 Ms is a bit less than fortnight.)

1.1.4
If a credential's lifetime can be extended by the user, using no more proof of identity than the credential itself, this is considered "renewal" of the credential, while if the process of extending the lifetime requires measures equivalent to those employed in its initial acquisition, we consider the result a new credential.

1.1.5
We specifically do not consider "post-dated" credentials -- those with lifetimes that begin at some point later than the time of the authentication act. Neither do we consider the relative strengths of cryptographic protocols, algorithms, and key lengths. We assume they are always designed, selected and implemented appropriately.

## 1.2. Identity

1.2.1
Sites will generally make authorization decisions on an aggregate basis, any Virtual Organization (VO).
However, at times it will be necessary to set access rights at the granularity of a single user. Sites must reserve the right, and preserve the ability, to set authorization at this level. Also, incident handling requires the ability to identify the legitimate owner of credentials presented during transactions under investigation.

Accordingly, every set of authentication credentials SHOULD be tied to the identity of an individual, because this provides stronger security by way of audit ability, revocation, and problem determination. However, there may be occasion to forfeit these benefits in order to provide temporary and generic identities.

For example, an Internet cafe could provide temporary (very limited lifetime) credentials authorizing use of grid resources based solely on the fact that access was purchased.  Such an identity may be as generic as "Customer 24."

Similar identities
- action traceable to a specific organization within a specific VO
- action traceable to a specific VO
- action purely anonymous

1.2.2
Secure anonymous communications may still be allowable, and appropriate, for functions that do not require user authentication.

For example, access to Grid resources is purchased and anonymous access if give; the user may still require secure conversation because the results of the data derived may have value.

## 1.3. Assurance

1.3.1.
An authentication system may provide multiple methods for a user to perform their initial authentication, and these methods may differ in their convenience, resistance to attack, and risks of exposure of secrets. Even when an implementation offers its users only one method, it may not be clear to relying parties which method it is.

Since some inverse correlation does exist between convenience and strength of authentication, there may be inducements to allow and employ multiple levels of authentication if sites make some class of services available through weaker but less burdensome authentication methods.

1.3.2.
We define three levels of authentication strength:

Strong - long-lived reusable secrets are not transmitted over the network.

Encrypted - long-lived reusable secrets are transmitted on the network in encrypted form. The encryption techniques (including key management) MUST be of sufficient strength that secrets are unlikely to be recovered by a hostile party before their expiration.

Cleartext - reusable identifying information (it would be an eageration to call it a secret) is transmitted in the clear. Cleartext authentication is considered equivalent in trength to no authentication at all.

1.3.4.
We recognize following modes of storage of users' long-term secrets, each with its own set of vulnerabilities:

1.3.4.1 What you know
Mental - secrets are held in users' own memory (PIN or password).

1.3.4.1 What you have
Secured - secrets are stored in electronic devices with credible protection against disclosure to unauthorized parties, even in the event of user carelessness.

Stored - secrets are stored in electronic devices in a manner that relies on users' willing diligence in protecting them against disclosure e.g. Biometric, or smartcard.

1.3.5.
It is not possible to give a strict ranking of storage modes discussed section 1.3.4 relative to safety without asking and answering a number of questions about the details of the secrets, their storage, and their registration as the users' authentication information. Also, users may perform unsafe actions (knowingly or unknowingly) which place their secrets at much greater risk of disclosure.

Authentication strength MUST be mechanically deducible from credentials. The method used to perform authentication SHOULD be deducible from credentials.

1.3.6.
There are a number of cases where processes running on a machine need to authenticate to other processes. Automated processes may have to act as authenticated clients and users may wish to have automatic software ("cron jobs") that require automatic authentication. All of these should be somehow restricted such that theft of credentials from an individual machine does not easily permit their reuse elsewhere. In either case, secrets will be of the "stored" class and must be considered to be stored in cleartext form, regardless of any measures which obfuscate them.

Authenticated identities of automated client processes SHOULD include identification of the machine which is intended to have access to the authentication secret.

Authentication methods based on stored secrets SHOULD indicate the machine from which they were used. If they do not, then this information MUST be available in auditable records.

## 1.4. Lifetimes

1.4.1.
All forms of digital credential in common use are subject to possible theft and misuse. The probability of such an event is monotonically nondecreasing with time. The countermeasures against eventual credential theft are expiration and revocation. Neither measure alone is sufficient to prevent all misuse, nor is the combination of the two.

1.4.1.1
User authentication credentials MUST NOT be valid for more than 1 Ms if there is no method for checking for revocation. User authentication credentials SHOULD be renewed or checked for revocation every 0.1 Ms.

1.4.1.2
Authorities issuing revocable credentials MUST publish the procedures for initiating revocation. In the case of X.509 certificates, each revocable certificate SHOULD include a pointer to such procedures. These procedures MUST include the loation and publication frequency of revocation information and an upper bound on the time required to act on a revocation request.

1.4.1.3
It SHOULD be possible for authority parties other than the credential issuer or the credential owner to initiate revocation, under some circumstances. For example the authority that vetted the identity of the user. The processing time bound above may not apply to third-party requests for revocation.

1.4.2.
The lifetime of authentication secrets is a separate parameter from the lifetime of credentials.

1.4.2.1
User secrets stored mentally SHOULD have a lifetime of 50 Ms or less. Some environments or applications may demand shorter lifetimes, down to perhaps 10 Ms.  These times may vary depending on the strength of the password enforced by the password requirements of the system.

1.4.2.2
Secured user secrets may reasonably have lifetimes of 100 Ms or more depending on the securing technology.

1.4.2.3
Stored user secrets SHOULD NOT be valid for more than 1 Ms, and if valid longer than that, their associated credentials MUST declare that fact.

1.4.2.4
The above lifetimes are relevant to both the strength of the password and the strength of the cypto-analysis or password

cracking tools. These lifetimes should be adjusted to reflect the current state of the art in these two related technologies.

# Chapter 2 Site Authorization Requirements

## 2.1. Authorization Requirements

### 2.1.1. Clarity of AuthZ

Assertions of membership's roles and groups in a VO MUST be able to be validated by relying parties. Currency of this information SHOULD not exceed 1Ms.

#### 2.1.1.2 The Resource Administrator Authorizes Groups and Roles

VO attributes describing the roles and groups MUST follow a published standard, agreed upon at least within the domain of the VO. This consistence gives the Authorizor or Resource Administrator a manageable and trusted view of the membership pool. The administrator MUST be able to trusts the concurrency of the roles and groups. This removes the need for Authorizer to have an understanding of each member. The Authorizer needs to only understand the groups and roles within this assigned membership pool.

#### 2.1.1.2.1

This standard MUST include:

- format of the attribute credential
- a definition of all attributes
- method of determining validity of credential
- method of determining expiration of credential

#### 2.1.1.3

VOs SHOULD provide a method providing membership and role/group information for a given user. An example of this might be extended attributes within the certificate.

### 2.1.2. Transparency of AuthZ information / policy

Certain groups or roles may require additional authorization before membership information is released (so as to not leak information about which accounts are privileged).

### 2.1.3. Protection of AuthZ info

Alterations of the information should only be possible through secure, authenticated access paths using procedures such that the sites are willing to trust the role / membership information returned. This requirement may involve a CP/CPS-like description for how virtual organizations maintain and protect this data.

Current proxy certificate specifications ensure that proxy and delegation operations never require private keys to be sent across the network. It is important to state clearly to developers that all future protocols must continue this practice. If it is necessary to send a passphrase or password across the network, they need to be encrypted at a strength equivalent to the strengh of the key.

### 2.1.4. Proxy Certificate Revalidation

Since the proxy certificates do not have any mechanism for being revoked, any proxy certificate must be revalidated every "standard day".

### 2.1.5. Authorization Level Dependent on Authentication Strength

The authorization for access to a resource at a particular level may depend on the strength of the authentication. The level of authentication must be included with the credential information presented to all resource managers.

2.1.6. Call-outs
Call-outs prior to access to resources MAY be provided as a form of authorization control by the virtual organization, the site(s) and each resource provider.

2.1.7. Re-authentication requires Re-authorization
If a credential expires and requires reauthentication, then reauthorization should also be performed by the Registration Authority (RA). The RA will vet the user's roles and groups. Resource specific authorizers will be made aware of any group or role changes, e.g. via the extended attributes of the certificate of the user.

2.1.8 Logging

Logs documenting the resource access decisions, policies, policy changes, and resource implementation of policies should be logged. The virtual organization, site(s) and resource managers and should log such events and retained these logs for 10**7 seconds (approximately 4 months). The logs must be protected to ensure privacy and integrity. The restrictions and safeguards should be published.

2.1.8.1 Logging requirements
- logs SHOULD be frequently archived on a machine different than the one they were generated
- at time of archival the logs SHOULD be digitally signed by the archival server
- if is recommended to constantly log grid access requests to a site wide logging server in addition to local logging at the resource, this increases auditability and protects from tampered logs if a machine has been compromized. (and maybe this needs to go into the accounting section, i.e. cross reference to section 3.4 )

2.2: Authorization requirements in Advanced Collaborative Environments

2.2.1 Authorization Policy Change Control

2.2.1.1 Authorization policies may change over time. Mechanisms to manage policy specification across the sphere of control of the resource, site, VO, application manager, and user should be provided.

2.2.1.2 A time delay between publication of a policy change and its' implementation or enforcement is to be expected. There should be prompt implementation of policy change. The resource manager will implement the policy change and log compliance. The resource manager will define a prompt and reasonable time delay appropriate for the resource.

2.2.1.3 Sites and virtual organizations must have the ability to suspend resource authorization for a particular grid identity without actually deleting the authorization and therefore possibly losing tracking information.

<REQUIRES_MORE_DISCUSSION_GGF8>

2.2.1.4 There MUST be the ability to quickly revoke a particular remote authorized service that may be operated under dubious procedures. For example, if a remote processing resource steals computation results, it should be removed from the directory of processing resources. This is difficult in the context of the current Grid Technology because of the open resource registration process and aggressive discovery algorithms. Similar such directory services on the Internet have a history of exploitation, such as the DNS recursive lookup hack.

<\REQUIRES_MORE_DISCUSSION_GGF8>

2.2.2 Granularity of Authorization

Depending on the application scenario the granularity requirement for authorization decisions vary from fine grain (based on individual subject, requested action and assets involved) to coarser grained authorization on the basis of groups or even sites. Support for role based access control mechanisms is specifically requested for future ACE systems.

2.3.

2.3. European Data Grid

2.3.1.
This section is based on the EU DataGrid "Security Requirements and Testbed 1 Security Implementation",
http://edms.cern.ch/document/340234/4.0

"We are largely basing authorization on the concept of the Virtual Organisation or VO. A Virtual Organisation is a collection of individuals and institutions that are defined according to a set of resource sharing rules. A Virtual Organisation is a dynamic collection of individuals, institutions and resources. Users will be members of one or more VO. Resources may belong to a VO, and allow their members access. VOs may collaborate, and allow one another use of their resources, possibly giving priority to their own members. (For example, there may be 2 Particle Physics experiments, funded by different sources. Each may set up their own VO. They may wish to set up their resources such that they can use one another's resources, but their own members get priority over their use. For instance, members of their own VO may submit jobs to a high priority queue, but members of the other VO may only submit jobs to a lower priority queue.) Within each Virtual Organisation, various roles are defined. A role is defined as an attribute of a principal that allows the principal to carry out certain actions." - Andrew McNab

2.3.2. Users or end entities may be members of any number of Virtual Organisations.

2.3.3.
Users or end entities may have any number of roles within a given Virtual Organisation

The Virtual Organisation must be able to decide user membership policy and user authorization policy.

2.3.4.
The owner of a resource or data MUST be able to allow or deny an end entity authorization to carry out an action based on any of the following:

1) By public access
2) By only having acceptable authentication
3) By membership of a VO
4) By role(s) within a VO
5) By membership of a combination of VOs and roles
6) By allowing selected certificates
7) Individual certificates may be banned

Note 2.3.4.1
We might want to get a clear picture up front what

the precedence rules are. I am assuming that the mostspecific overrule the less specific or something like that, but with 8 different ways, the calculations could get a little funny.

note 2.4.7.1.2
The ability to quickly revoke a particular remote authz service (like a VO authz service someplace) is something to think about. Once we're stuck trusting remote servers that may be operated under dubious procedures, it will be important that when a compromise is detected, we can quickly lock them out.

2.3.5
The authorization method must allow any combination of the above authorization requirements, including any combination of VOs and roles

2.3.6
It should be possible to base authorization on any one of the following, in addition to the authorization requirements above:

1) Data name (Any of file, directory etc.)
2) Storage element name (= fileserver)
3) Operation (including metadata and file operations)
4) Resource usage limits. (E.g. quota)

2.4 fine grain access control policies

2.4.1
There MUST be no restrictions on the degree/level of granularity of authorization. In particular, no hard-coded limits to how the granularity is set should exist. This should include, for example, allowing authorization to a hierarchy of directories, individual directories, or individual files. It may become burdensome on the resource to support high level of granularity, therefore it is left to the resource to set a practical level of granularity.

2.4.2
It MUST be possible to determine the list of resources to which an end entity has access and what actions they are allowed to carry out in the VO(s) and role(s) set for the current session. The burden of creating this list is on the end entity. It is left to the end entity to know or lookup or discover the resource and query for access permissions. This relieves the resource from having to know how to report to the end entities. This also averts a security vulnerability similar to the historical NIS (Network Information Services) hack in which the complete access lists being pushed to slave servers was intercepted and exploited.

2.4.3
It MUST be possible to determine if a role or group has access to a resource. This access information is necessary to accurately stage and schedule jobs. This access information is sensitive because it could be used to exploit the Grids security. For example, knowing that Bob has access to the targeted resource, the hackers attention is turned to Bob or his home computer. Therefore the resouces access information MUST be known in its' complete form to the resource administrator and Grid security personnel for security audit and forensic purposes. Others MUST have access to authorization data ONLY in the form
1) permit and permit qualifier (e.g. PERMIT/always or PERMIT/8:00am-5:00pm)
2) or denied and denied qualifier (e.g. DENY/always or DENY/QoS load).
This information MUST pertain only to their identity.

2.4.3.1
There is a dynamic nature to authorized access in that it may depend on the resource load, quality of service, or time of day. If authorization access changes during access, an error code SHOULD be propagated back to the application or the application SHOULD query for the authorization deny qualifier.

2.5 Transparency

2.5.1
The authorization method must be application independent.

2.5.2 The authorization process must be the same and consistent within a VO. Implementations of this process could differ as long as the same process or procedures are followed.

2.5.3 The complexities of different levels of authorization and allowing some entities access to one portion of data and disallowing access to another portion of related data is not new or specific to the Grid computing. For example, patient information the doctor sees may be unethical for the insurance company to view. An entire standard body is addressing these issues (Liberty Alliance www.projectliberty.org).

2.5.6 It is necessary that controls are in place to allow the implementation of these authorization decisions and that these controls are not overly complex as to tax the abilities of the resource administrator.

2.5.7 The consistance and transparency to the application is aided by the use of standardized error codes of authorization denials. The error information MUST not provide more information than necessary as to create a security risk. An error return code MAY be accompanied with a log entry number to assist the resource adminitrator in synchronizing the denial instance. For example, a user may call a help desk to report access problems, giving the error code and log entry number. The resource administer can reference this log entry number to provide detailed information which may not be suitable or others to view.

2.5.8
My main concerns revolve around:

1) An authz system that is understandable for VO's and site resource providers. I think that comprehensibility _must_ trump generality and power. An authz system that is too complicated is a liability - we can't afford an authz system where you don't know why someone can or can't get access. Even with something as dumb as Unix file permissions, people get stuck.

2) How does the system perform under the kind of failure modes we commonly worry about (network partition, server compromise, etc...)

2.6. Authorization Servers

2.6.1 There should be authorization servers SHOULD be localized to the service. Grids SHOULD gracefully survive partitioning so that local services can continue their operation in case a resource is disconnected or to avoid a DoS attack. This may required redundant or distributed Authorization Services.

2.7 Authorization Revocation

2.7.1

It must be possible to disable a user's authorization in the following ways:
1) It must be possible to remove a user from a VO.
2) It must be possible to remove a given role or a number of roles from a given user.

2.7.2
This should be done in a time frame consistent to the authentication revocation of 0.1Ms.

2.7.3
After the user has authenticated himself, the user must be able to select and de-select VOs and roles. This is analogous to the substitute user or 'su' command on UNIX systems. This allow an entit to change a role briefly for a critical section before returning to a role and access less vulnerable or potentially dangerous. This consistent with section 2.1.2.

<REQUIRES_MORE_DISCUSSION_GGF8>

2.7.4

Is there a need for mutual authorization. An application or end entity may need assurances that the recourse is authorized to run a specific job. The distributed program or grid job in and of itself may be of value. The results may be of value and need protection from dubious or resources with poor security.

2.7.4.1 A grid job may need to specify that it is only run on systems with security level B operating systems, or systems not directly connected to the internet, or system with OS's prone to hacking. This is more relevant in the OGSA model where service factories may incorporate more resouces to handle service request loads.

<\REQUIRES_MORE_DISCUSSION_GGF8>

2.8 The user has power to choose current role, and current set of privileges.

2.8.1
The authorization requirements on data access should hold regardless of replication.

2.9 Maintain Policy

The Authorization mechanism must preserve the identity of the user, i.e. the DN or distinguished name of the user.

It should be possible to assign a user to set the authorization on data access.

2.10 Delegation of Authority

2.10.1.
If files are replicated, authorization for access to this replicated data must not depend on one other single site being available.

2.10.1.1
This key term is "replicated". Data is replicated to provide a higher level of availability. This availability would be compromised if the the authorization was dependent on the origin site's authorization server.

2.10.1.2
There is an inherent trust in the Delegation of Authority model. For example, one is authorized to music CD when the set price is met. I

subsequent delegation to others to replicate the music CD is prohibited in the Term and Conditions.

2.11 Role Confirmation

2.11.1
It must be possible to confirm that a user has the VO membership(s) and Role(s) they are claiming at the time they request an action. However, However, it must NOT be possible to produce a list of members of a VO, or which VOs a user belongs to.

Note 2.11.1.1
privacy,  protection of authorization  policies But MUST NOT be possible seems too strict, only e.g. an VO admin  should be allowed to create such a list but why NOT possible

The VO should be able to specify a list of either which specific resources, or which specific VO's resources are acceptable when a user is in a particular role.

2.12.1 Providing credentials to service
The auth and authz creds that a user presents should be made available to the execution environment of something like a gatekeeper job manager. In other words, the gatekeeper may have passed you through with your creds, but if this means you are running a job, the auth/authz creds should be made available via some mechanism like environment variables.

Chapter 3  Site Accounting Requirements

3.1 Accounting Requirements Introduction

Accounting is important beyond charging or purchasing resources. Accounting links to other business IT processes such as business planning, return on assets and management information.
Accounting is importance is beyond accurate billing. IT use accounting for controlling and managing operational costs. Accounting links to other IT disciplines such capacity planning, service level management, performance management.

Accounting has historically had close ties to Authentication and Authorization because of the certainty in which they identify the entity to be associated with the accounting data. This is particularly important in the areas of security audits, intrusion detection, and computer and network forensics. Recall the book "Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage" by Clifford Stoll who spoke at GGF6; this true tale all started with an accounting error. I might add, this document reads with the same excitement as "Cuckoo's Egg"; assuming you are on the appropriate medications.

3.2 Document Goals

The focus of this document is divided into two categories: Grid Resource Accounting, and Grid AAAccounting
Grid AAAccounting is the focus on accounting as a security component, and the need for a seamless relationship between Accounting the Authentication and Authorization components of the Grid.  Simply put, with a small addition to

existing accounting data, a AAAccounting mechanism could greatly enhance Grid security.

Grid Resource Accounting is the more traditional sense of accounting that accounts for resources usage and billing.

3.2.2 Requirements Gathering for Grid AAAccounting

Requirements for Grid AAAccounting focuses on the relationship of monitoring and metering authentication and authorization for security purposes. This information binds an end entity to the resource being access for the time and duration of access. The consumer of this information is Grid admin, helpdesk , intrusion detection or computer forensics.

3.2.3 Requirements Gathering for Grid Resource Accounting

It is important to understand how the accounting data will be used. This will help define the accounting data gathered and the data flow.
It is the goal of this document to describe the requirements of Grid accounting components which satisfy a broad range of instances and usage. This document will also identify other current Grid working groups and accounting standards that are addressing these needs.

3.3 Non-Goals

This document will understand the consumers of the accounting data and their requirements, but will not analyze the consumers or make recommendations on how consumers should process the accounting data.
It is not the goal of this document to reproduce or reinvent past accounting standards or duplicate current Grid accounting work.
Relationship with Authentication and Authorization
Accounting has historically had close ties to Authentication and Authorization because of the certainty in which they identify the entity to be associated with the accounting data.

3.4 Grid AAAccounting

The Grid AAAccounting examines accounting requirements from a security perspective: audit logs, intrusion detection, and forensics. These requirements are not disjoint for mainstream accounting concerned with billing and metering, but in this section the requirements are described from the security perspective.

3.4.1
Grid AAAccounting must monitor or log the following data per resource access.
   Resource
   End Entity Identity and Provenance
   Authentication and Authorization
   Action Time and Duration

3.4.2 Resource Identification RID

The resource must be identified.
The resource identity can be layered or accumulative or onion fashioned. This identification may be any or all of the following and more:
   1) IP address
   2) Web Service
   3) vnode, or inode and generation or some other file handle
   4) file set or disk volume group

The RID should be descriptive of the state of the resource.
For example, if the resource is a file, the exact content of the file at the

time of access would be an optimal piece of information for a forensic analogy. This type of metadata is difficult and expensive to maintain, and usually requires replay logs for the most accurate view of the data at and during the time of access. None the less, the more accurate the accounting description of the resource the better the assessment and recovery from a hack is possible.

### 3.4.3 End Entity Identification EEID

The EEID accurately describes the end entity of the resource. Commonly this will be a GSI proxy certificate, which can be easily translated into an certificate and person to whom that certificate was issued.

### 3.4.4 Globus Toolkit 2.* EEID

It is appropriate to talk specifically about the Globus toolkit because it is a widely used Grid technology and moreover illustrates a relationship that is common among Grid technologies.

#### 3.4.4.1 Relationship between EEID and Process id

Intrusion detection at a file system level when triggered identifies the PID (process id) of the offender. Via the system process table, the associated UID (user id) and PPID (parent process ID) can easily be identified. When a Grid job is submitted and runs on a Grid resource, the parent process is the UID mapped to the certificate in /etc/grid-security/grid-mapfile during the authorization process. Many certificates may be mapped to the same UID. This masks an audit trail needed to link all of the connections from the offending process to the EEID.

#### 3.4.4.2

The two crucial pieces of information are the PID of the process running on the Grid resource and the EEID responsible for initiating this process. Both the PID and the EEID are known but not necessarily recorded consistently or together. The globus-gatekeeper will log the EEID at authentication time in the syslogd data.

For example

```
Feb 14 09:31:32 ipsec GRAM gatekeeper[29452]: Authenticated globus user:
/C=US/O=IBM/OU=GridLPP/OU=austin.ibm.com/CN=shawnm
```

In this example, the EEID can easily be tracked via the CA and RA back to a singular user. The disjoint occurs with the recording of the PID of the actual process that is run on behalf of the EEID on the Grid resource. The PID is returned to the initiator in the form of a JobID.
For example

```
<274> globus-job-submit ipsec /bin/ls ls /tmp
https://ipsec.austin.ibm.com:62960/27126/1045236692/
```

#### 3.4.4.3

The middle number is the PID of the ‘ls' command run on the Grid resource ipsec.austin.ibm.com. The JobID, which contains the PID, and the EEID should be sent as part of the Grid AAAccounting monitor data. This data should not be recorded locally because it allows a hacker a means to cover his tracks. All Grid AAAccounting data should be reported to a remote central system.
The provenance of the process or job must extend to the true origin. The GIS model allows for the propagation of jobs and the inheritance of security credentials. Simply put, as a job propagates from Grid resource to Grid resource, EEID must remain consistent or any transition of identity must be monitor. Perhaps stepping beyond the scope of this document but for the purposes of illustrating a point, it is obvious that if a process inherits credentials beyond the subset of its' current credentials, an alarm should be triggered.

3.4.5 Authentication and Authorization

Knowing the provenance of a job should allow the audit trail to quickly discern the authentication and authorization used to gain access to the Grid. Again. Back to the example of the Globus Toolkit.

The EEID or proxy certificate is logged by gatekeeper on the Grid resource. This is a logging of the authorization processes. The actual authentication took place on the provenance node with grid-proxy-init when the passphrase was entered and the proxy certificate created. The authentication process should be monitored. Currently it is not possible to distinguish between a valid authentication via grid-proxy-init and the stealing of the proxy certificate out of /tmp.

This is analogous to the "su" command (substitute user) which is logged by syslog and in sulog. When the grid-proxy-init command is issued the user is talking on the identity of a particular Grid user. This information should be part of the Grid AAAccounting data.

3.4.6 Action, Time and Duration

This section will have some intermingling of the accounting requirement as they relate to security and as they relate to On Demand computing. This is done to illustrate that the exact same accounting data is used for two very different purposes.

3.4.6.1 Attempted Action relating to IDS and OGSA

The action of the process running on the Grid resource should be part of the Grid AAAccounting data. The action of the process may be attempted but unsuccessful or denied. An example failed su attempts or failed logins. Action attempts are critical for behavior based components of Intrusion Detection Systems (IDS).

Additionally, failed actions may be a consequence of a resource shortage or outage. For example, in the Open Grid Service Architecture (OGSA) model this information could be used to create an additional service factory.

3.4.6.2 Time and Duration Relating to IDS and On Demand Computing

The time and duration of the action are critical to computer forensics, as they report on who was in the candy store and for how long. This allows for the creation of a time line of activity. Action, time and duration are equally important to both intrusion detection, On Demand or dynamic services, and autonomic or self healing services.

3.4.7 Grid AAAccounting Conclusion

In a Grid environment it is important to monitor a causally connected sequence of events. It is important to be able to traverse this sequence of events from authentication to action taken on the remote resource. The proper accounting data can enable intrusion detection, the detection of malicious behavior and provide security audit trail.

3.5 Requirements Gathering for Grid Resource Accounting

I do not view this as an abdication of responsibility to leave this section to other GGF working groups. I view this as efficient means of coordination between different GGF groups. I believe the Grid AAAccounting is closely affiliated with security, where as the more traditional computer accounting belongs more in the area of GGF-SRM, Scheduling and Resource Management Area, (http://www.gridforum.org/3_SRM/srm.h) Specifically in GGF Resource Usage Service Working Group

GASX Grid Service Accounting Extensions by Anthony Beardsmore
Extreme Blue Grid Accounting Project (Grid Service Accounting Extensions) by James Magowan.

### 3.6 Existing standards and practices

#### 3.6.1 Accounting Institutes

I have not been able to find any standards for computing or IT accounting relating to traditional CPA accounting or from other standard bodies such as Oasis or Liberty Alliance.

IETF

Existing work has been done in this area but not necessarily relating to Grid in the a set of IETF RFCs.
The Related RFCs are:

RFC3127
Authentication, Authorization, and Accounting: Protocol Evaluation

RFC2989
Criteria for Evaluating AAA Protocols for Network Access

RFC2977
Mobile IP Authentication, Authorization, and Accounting Requirements

RFC2975
Introduction to Accounting Management

RFC2906
AAA Authorization Requirements

RFC2905
AAA Authorization Application

RFC2904
AAA Authorization Framework

RFC2903
Generic AAA Architecture

RFC2866
RADIUS Accounting

IETF Draft on DIAMETER BASE Protocol
<draft-ietf-aaa-diameter-17.txt>
<http://www.ietf.org/html.charters/aaa-charter.html>

Of the RFCs, I found the the RADIUS Accounting standard to be the most interesting because the nature of securely logging onto a network via RADIUS, is similar to the nature of securely logging onto a Grid. There is considerable work in this standard that can be leveraged in implementing a Grid Accounting standard.

Footnotes
[1] GGF Grid Certificayte Policy - WG paper "CA-Based Trust Model of Grid Authentication"

Glossary
resource administrator - the owner of a resource or the entity having control of the resource acting as an extension of the owner.

resource authorities - synonym for resource administrator
Grid security personnel - trusted personnel tasked with ensuring the integrity of the Grids security

end entity - an identifiable user, or service using resources.

user - synonym for end entity

At the recent PKI Workshop I saw the following names used

Subject Authorities - issue attributes to subjects (the users)

Policy Authorities - issue more general (than resource specific) policies (e.g. site policies)

Environment Authorities - issue statements about environmental points

## 4.0 Intellectual Property Statement

The GGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the GGF Secretariat.

The GGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the GGF Executive Director.

## Full Copyright Notice